

TD 14 : Initiation au HTML/CSS

Le but de ce TD est de comprendre comment sont écrites les pages Web basiques, aussi appelées pages statiques (Web 1.0). Une telle page Web contient deux parties :

1. **HTML** : le fichier HTML contient la structure de la page et son contenu ; en plus du texte brut, il donne du sens au texte en indiquant ce qui relève d'un paragraphe, d'un titre, *etc.*, à l'aide de balises (exemple `<p>`, `<title>`...) ;
2. **CSS** : le fichier CSS est responsable de la mise en page de ces éléments (mettre ce paragraphe en rose, utiliser la fonte « Sans Serif » pour ce titre...).

Partie 1 : HTML

Le navigateur (Firefox, Chrome, Safari, IE/Edge...) est le logiciel qui nous permet de visualiser les pages Web. Le but de ce TD est de démystifier la façon dont sont interprétés ces deux types de fichiers par le navigateur. Pour cela nous allons réaliser un site dont le rendu correspond au fichier **target.png**, en partant du fichier **index.txt**, qui contient le contenu quasiment « brut » du site à réaliser.

L'ensemble des fichiers se trouve dans le serveur classe.

Pour créer une page Web, il suffit de créer un fichier texte et de lui donner l'extension .html pour que le navigateur comprenne qu'il doit l'interpréter comme un document HTML.

- Sauvegardez le fichier **index.txt** en local dans un dossier HTMLCSS/TD13/.
- Sauvegardez le même fichier **index.txt** en local dans le dossier HTMLCSS/TD13/ en le **renommant** **index.html**.
- Ouvrez les deux fichiers **index.txt** et **index.html** dans le navigateur.
- Quelles différences d'affichage observez-vous ? Pourquoi ?
- Comment sont affichés des sauts de ligne d'un document HTML ?
- Comment est affiché un texte HTML entouré de `<!--` et `-->` ?

Le HTML, qui signifie *HyperText Markup Language* (langage de balisage d'hypertexte), est donc un langage à balises contenant des liens, dits *hypertextes*, vers d'autres documents.

Le HTML est un standard, c'est-à-dire un langage complètement décrit (n'hésitez pas à jeter un rapide coup d'œil à sa **spécification**, un document très technique mais complet).

Testons la conformité de **index.html au standard HTML5 à l'aide du validateur <https://html5.validator.nu/>. Quelles sont les erreurs indiquées ?**

Commençons par l'erreur

Error: Non-space characters found without seeing a doctype first. Expected `<!DOCTYPE html>`.

Le validateur nous dit qu'il s'attendait à voir `<!DOCTYPE html>` au début de notre document. Cette balise sert à déclarer que le document est écrit en HTML5. En effet, il existe plusieurs standards de « langages HTML » : HTML4, XHTML, HTML5... Aujourd'hui, les gens utilisent majoritairement HTML5 et nous ferons de même.

Pour que le document soit valide et reconnu comme un document HTML 5, **ajoutez** la balise `<!DOCTYPE html>` au tout début du fichier.

Retestez la conformité de votre document.

Le validateur nous indique **The character encoding was not declared.**

L'encodage indique la façon dont le fichier est enregistré (UTF-8, ANSI, iso-8859-15...). Si la plupart des caractères sont enregistrés de manière standard, les caractères spéciaux (accents, œ...) peuvent être enregistrés de manières très différentes.

Lorsque l'encodage n'est pas spécifié, le navigateur risque d'afficher Å© au lieu de é à cause d'une mauvaise détection de l'encodage. En effet, le caractère é est enregistré en UTF-8 de la même manière que Å© est enregistré en iso-8859-15 (encodage encore très utilisé dans Windows).

Spécifier l'encodage des caractères est donc nécessaire pour que les caractères spéciaux de votre page soient bien affichés. Nous utiliserons toujours l'encodage UTF-8 (et idéalement tout le monde devrait aussi l'utiliser).

Ajoutez donc la ligne suivante qui déclare l'encodage dans l'en-tête du document juste après le DOCTYPE.

```
<meta charset="utf-8">
```

Rouvrez `index.html` dans le navigateur et vérifiez que tous les accents s'affichent bien.

Note : si les accents ne marchent toujours pas, c'est parce qu'il **ne suffit pas** de dire que votre fichier est en UTF-8. Il faut aussi que votre fichier soit **bien enregistré en UTF-8**. Par exemple dans Notepad++, l'encodage est marqué tout en bas à droite. Vous pouvez convertir votre fichier en UTF-8 en cherchant dans les menus.

Retestez la conformité de votre document.

```
<title>Le site non officiel de Chuck Norris</title>
```

À ce stade, le validateur indique que le fichier `index.html` est un document HTML5 valide.

Nous avons vu dans l'exercice précédent nos premières balises `<meta>` et `<title>`. Les balises permettent de structurer le document. Elles annotent le texte qu'elles contiennent et permettent donc d'y ajouter du sens. On distingue deux types de balises :

- La plupart des balises **englobent du contenu** : elles commencent par une *balise ouvrante* `<mabalise>`, puis le *contenu* que l'on veut « annoter » et finissent par une *balise fermante* `</mabalise>`.
Par exemple, la balise `<title>` sert à dire que le texte qu'elle englobe sera le titre du document ;
- **Certaines balises n'acceptent pas de contenu** : elles consistent juste d'une balise ouvrante.
Par exemple, nous avons vu la balise `<meta>` et nous verrons bientôt `<img>`, `<br>`...

Autres exemples :

```
<p>Ceci est un paragraphe HTML puisqu'il est entouré des balises 'p' </p>
```

```
La balise 'br' du saut de ligne ne prend pas de contenu <br>
```

Servons-nous donc des balises pour créer une bonne structure de base de document HTML :

```
<!DOCTYPE html>

<html>
  <head>
    <!-- L'en-tête du document avec au moins un titre -->
    <title>Un titre qui s'affiche dans l'onglet du navigateur</title>
  </head>
  <body>
    <!-- Le corps du document -->
  </body>
</html>
```

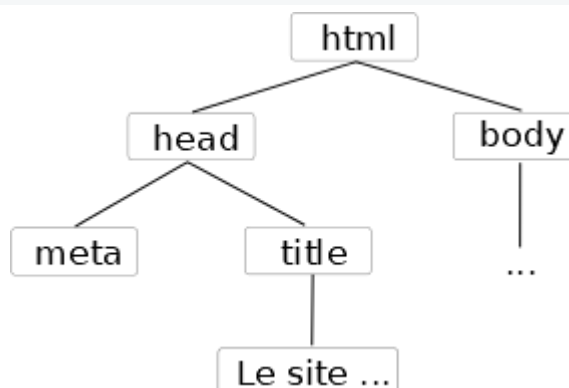
Après la ligne `<!DOCTYPE html>` de déclaration du langage, le document est inclus dans la balise `<html>` et est composé de deux parties :

- l'en-tête `<head>` contient des informations sur le document HTML, comme
 - un `<title>` (balise obligatoire),
 - un `<meta>` pour définir l'encodage ;
- le corps `<body>` contient le vrai contenu. Nous verrons des exemples de balises dans la section *Balises communes*.

Les balises HTML donnent une structure d'arbre au document. Dans notre exemple `index.html`

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Le site non officiel de Chuck Norris</title>
  </head>
  <body>
    ...
  </body>
</html>
```

l'arbre est le suivant :



- `<html>` est l'élément racine.
- `<head>` et `<body>` sont les deux fils de l'élément `<html>`.
- `<title>` et `<meta>` sont deux fils de l'élément `<head>`.
- « Le site non officiel de Chuck Norris » est un fils de l'élément `<title>`.

Mettez à jour votre page `index.html` pour qu'elle respecte la structure HTML ci-dessus. (Vous devez ajouter les balises `<html>`, `<head>` et `<body>`.)

Retestez la conformité de votre document.

Dans la suite du TD, nous allons utiliser notre navigateur pour « inspecter » notre page Internet. Pour cela nous conseillons Chrome ou Firefox. Appuyer sur la touche F12. Les outils de développement affichent deux parties bien distinctes, une dédiée au HTML et l'autre... aux CSS. Ces outils sont fabuleux pour apprendre comment se construit une page Internet.

Il y a trois façons de s'intéresser à un élément en particulier :

- un clic droit avec la souris dans la page affichée, suivi d'un « Inspecter/Examiner l'élément », permet de voir le code HTML correspondant dans l'outil de développement ;
- un clic sur  dans l'outil de développement permet d'aller inspecter une zone d'intérêt dans la page (par survol avec la souris) ;
- quand on passe la souris au-dessus d'un élément dans l'outil de développement, il le colore dans la page.

➤ Familiarisez-vous avec ces trois techniques en inspectant la page du TD.

Par exemple faites un clic droit sur l'élément « il y a trois façons... » puis « inspecter l'élément ». L'outil de développement doit vous présenter le code HTML et vous positionner directement sur `<p>Il y a ...`.

Il est possible d'ajouter des commentaires dans le HTML. Ces commentaires ne sont pas interprétés par le navigateur, et ne sont donc pas affichés (mais ils restent présents dans le code source). Il s'agit donc d'informations laissées par des développeurs pour des développeurs. On les place entre les balises `<!--` et `-->` :

```
<!--Cela est un commentaire dans un fichier HTML -->
```

Il y a justement des commentaires dans le fichier `index.html`, comme autant de consignes afin de les remplacer par du HTML. Nous expliciterons ces dernières dans les sections suivantes.

Nous allons commencer par ajouter de la structure à notre page. Pour ce faire nous allons utiliser les balises à contenu `<h1>` à `<h6>` pour identifier les différentes sections :

- `<h1>` est utilisé pour les gros titres du document ;
- `<h2>` est utilisé pour les sections du document ;
- `<h3>` est utilisé pour les sous-sections du document et ainsi de suite.

Par exemple, le titre ci-dessus est obtenu avec le code suivant :

```
<h2>Titres</h2>
```

- Vérifiez que le titre **Titres** juste au-dessus est bien un `<h2>` à l'aide des outils de développement en faisant un clic droit dessus.
- Ajoutez la balise `<h2>` aux éléments de `index.html` marqués par les commentaires : `<!--section -->`.
- Ajoutez la balise `<h3>` aux éléments de `index.html` marqués par les commentaires : `<!--sous section -->`.

Retestez la conformité de votre document.

- Utilisez maintenant les balises `<p>` et `</p>` autour des paragraphes du document. Les paragraphes vous sont signifiés par `<!--début paragraphe -->` et `<!--fin paragraphe -->`.

Note : si vous faites un clic droit suivi de « inspecter l'élément » sur ce paragraphe, vous verrez justement que ce texte est dans un paragraphe.

En HTML nous pouvons faire des listes ordonnées (numérotées) ou non ordonnées :

```
<ul> <!-- ul pour unordered list -->
  <li>premier item non ordonné</li> <!-- li pour list item -->
  <li>deuxième item</li>
</ul>
<ol> <!-- ol pour ordered list -->
  <li>premier item ordonné</li>
  <li>deuxième item</li>
</ol>
```

Ce qui donne une fois interprété par le moteur de rendu du navigateur :

- premier item non ordonné
 - deuxième item
- 1. premier item ordonné
- 2. deuxième item

Utilisez les balises `<ul>` et `<li>` pour structurer la liste à puces `<!--liste -->` dans `index.html`. (Ne vous souciez pas encore des commentaires `<!-- lien externe -->`)

Utiliser les balises `<ol>` et `<li>` pour structurer la liste numérotée `<!--liste numérotée -->` dans `index.html`.

Retestez la conformité de votre document. (En fait vous devriez le tester régulièrement de vous-même, mais on vous tient par la main pour ce premier TD).

Pour insérer une image, on peut utiliser la balise

```

```

Cette balise n'a pas de balise fermante, car elle ne peut avoir de contenu (cf. **le début du TD**). On remarque qu'elle possède deux champs `src` et `alt` que l'on appelle les **attributs** de la balise. Les attributs se trouvent **toujours** dans la balise ouvrante.

L'attribut `src` doit contenir l'adresse de l'image. L'attribut `alt` permet d'ajouter un texte alternatif pour les navigateurs ne pouvant les afficher (navigateur textuel **Lynx**) ou pour les personnes ne pouvant pas bien les distinguer (aveugles ou déficits visuels légers). **Attention**, l'attribut `alt` est obligatoire.

Enregistrez l'image **chuck-jeune.jpg** dans un répertoire `images` par rapport à votre fichier `index.html`.

Remplacez le commentaire `<!--l'image de Chuck Young doit être positionnée ici -->` par la balise `<img>` suivante

```

```

Note : l'adresse `./images/chuck-jeune.jpg` est une adresse relative. Le point signifie « le dossier de la page Web courante ». Donc on va aller chercher l'image dans le dossier `images` du répertoire contenant la page Web `index.html`.

Faites de même avec l'image **beware.jpg** à positionner en lieu et place du commentaire `<!-- l'image de Chuck Beware ici -->`.

Devinez quoi ? Il faut retester la conformité de votre document.

L'un des éléments les plus emblématiques du HTML est sans doute la balise `<a>`. Elle permet de faire des liens hypertextes (le HT dans HTML).

Un lien est composé principalement d'une URL cible et d'un libellé (le texte cliquable souvent souligné en bleu) :

```
<a href="http://urlcible">le libellé</a>
```

On peut renseigner l'URL complète de la cible (URL en chemin absolu), par exemple :

```
<a href="http://lynx.browser.org/">Lynx</a>
```

ou donner une adresse relative à la page courante (URL en chemin relatif), par exemple :

```
<a href="./images/chuck-jeune.png">Image</a>
```

Remplacez les commentaires `<!-- lien externe ...` par des balises `<a>` avec la bonne adresse.

Testez la conformité...

On peut ajouter à ces *liens externes* une partie *lien interne* basée sur les ancres `#monancree`. Toutes les balises peuvent prendre un attribut `id` comme dans l'exemple suivant. **Attention**, la valeur de cet attribut doit être **unique** dans le document.

```
<h2 id="un_identifiant">
```

On peut alors faire un lien vers cette balise en ajoutant `#un_identifiant` à la fin de l'URL. Par exemple, voici un lien vers la balise d'identifiant `un_identifiant` interne à la page Web courante `index.html` :

```
<a href="index.html#un_identifiant">Exemple de lien interne</a>
```

On peut raccourcir l'URL comme suit. Si on n'indique pas le document, c'est que l'on pointe vers le document courant par défaut.

```
<a href="#un_identifiant">Exemple de lien interne</a>
```

- Remplacez le commentaire `<!-- lien interne ...-->` de `index.html` par une balise `<a>` qui pointera sur l'une des premières balises. Vous aurez donc besoin d'ajouter un identifiant à cette balise cible.
- Testez vos liens en cliquant dessus. Ils doivent vous emmener sur la balise dont l'identifiant correspond.
- **Testez** la conformité...

La balise `<em>` permet de mettre en évidence des passages importants dans un texte.

Justement, il faut mettre en exergue le fait que Chuck Norris est très fort dans différents arts martiaux. Pour cela il faut mettre en emphase la phrase qui suit le commentaire : `<!-- mettre en emphase cette phrase -->` dans le fichier index.html.

Note : il existe un autre type d'emphase qui s'obtient avec la balise `<strong>`.

Voici un magnifique exemple de citation :

Un biscuit ça n'a pas de 'spirit', c'est juste un biscuit. Mais avant c'était du lait, des œufs. Et dans les œufs, il y a la vie potentielle.

Jean-Claude Van Damme

Les citations permettent d'identifier un court texte sur lequel on veut attirer l'attention. Cela est utilisé notamment pour montrer qu'on a du 'spirit'.

- Allez voir le code source de notre citation à l'aide des outils de développement. Quelles sont les deux **nouvelles** balises utilisées ?
- La première balise (qui commence par un `b`) entoure la citation complète tandis que la deuxième (qui commence par un `c`) contient uniquement la référence (l'auteur, le livre...).
- Utilisez ces deux balises pour mettre en avant la citation en tout début de document (recherchez `<!-- utiliser blockquote ici -->`).

Partie 2 : CSS

Les standards définissant le CSS sont publiés par le World Wide Web Consortium (**W3C**) à l'adresse <http://www.w3.org/Style/CSS/>. Cascading Style Sheets (CSS) est un mécanisme simple pour ajouter du style (exemple fonte, couleurs, espace) à un document web.

Le CSS est responsable du rendu du site sur votre écran, mais aussi sur smartphone et des impressions papier (des ensembles de règles CSS peuvent être spécifiées pour chacun de ces médias).

Savoir les bases du CSS est relativement facile et indispensable pour qui veut travailler dans les métiers du Web. En maîtriser tous les aspects est un métier (celui d'intégrateur Web, qui traduit en HTML et CSS le travail du Web-designer).

Voici un exemple de règle CSS

```
h3 {  
  font-style:italic;  
  color:blue;  
}
```

Cette règle CSS donne à toutes les balises `<h3>` le style italique et définit comme couleur de texte le bleu. Une règle CSS est composée de deux parties :

- un sélecteur CSS qui indique à quels éléments HTML s'applique le style. Dans notre exemple, le sélecteur `h3` signifie que nous allons appliquer le style suivant à toutes les balises HTML `<h3>` ;
- un bloc de déclarations composé de plusieurs paires propriété CSS/valeur. Dans notre exemple, le bloc contient deux déclarations :

- Nous donnons la valeur `italic` à la propriété CSS `font-style`,
- nous donnons la valeur `blue` à la propriété CSS `color`.

Mettons en place nos règles CSS :

1. Nous allons travailler principalement sur ce TD sur un fichier `styles.css`. Créer un fichier texte vide `styles.css` dans un nouveau répertoire `css/`. Nous déclarerons dans ce fichier des règles CSS ;
2. Copiez dans `styles.css` le style ci-dessus `h3 { ... }`.

Ouvrez la page `index.html` et remarquez que le style ne s'est pas appliqué, car nous n'avons pas encore lié le HTML avec le CSS ;

3. Dans le fichier `index.html`, ajoutez la ligne suivante dans l'en-tête du document HTML (dans la partie `head`) :

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

Cette ligne permet de charger le fichier de style CSS `styles.css` et de l'appliquer à la page Web ;

4. Rouvrez la page `index.html` et notez que le style est désormais appliqué.

Nous allons vous présenter quelques propriétés CSS et la façon de concevoir des sélecteurs.

Pour la partie HTML, les outils de développement étaient déjà vos amis ; pour le CSS, ils sont promus au grade de « best-friend-forever ». Sélectionner un élément HTML avec les outils de développement ne permet pas seulement de voir les règles CSS appliquées à ce dernier, il permet aussi de les **changer**. Autant dire qu'il est conseillé d'abuser de cet outil pendant le TD pour bidouiller tout et n'importe quoi.

5. Inspectez avec les outils de développement (F12) l'un des textes en rouge. Dans la colonne de droite, vous voyez les styles CSS appliqués à cette balise.
6. Désactivez la déclaration `color:#d00` et remarquez le changement.
7. Remarquez que vos changements disparaissent quand vous rafraîchissez la page (F5). En effet, il faut les reporter dans la feuille de style pour les sauvegarder.

En CSS, seuls les commentaires avec `/*` et `*/` sont autorisés. Si vous utilisez `//` dans votre fichier `styles.css` vous allez avoir des problèmes (les règles CSS suivantes ne seront pas appliquées).

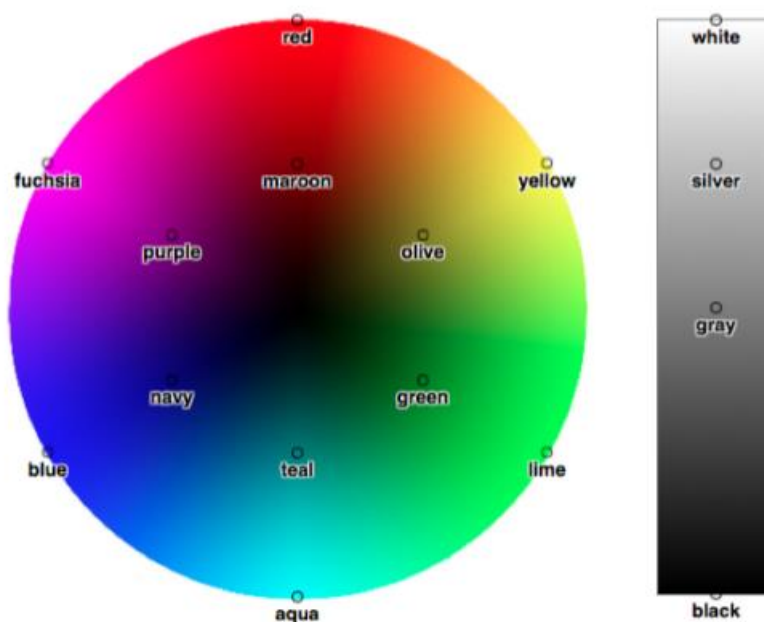
8. Commentez la règle CSS `h3 { ... }` dans `styles.css` et remarquez que les titres `<h3>` ne sont plus en bleu ni en italique.

Comme dit précédemment, les blocs de déclarations comportent plusieurs déclarations de la forme `proprieteCSS:valeur`. Nous présentons ici quelques exemples de propriétés CSS et de valeurs associées.

Les couleurs peuvent s'utiliser sur plusieurs attributs d'un élément HTML :

- la couleur du texte : `color:red` ;
- la couleur du fond : `background-color:blue` ;
- la couleur de la bordure : `border-color:yellow` ;...

Les 16 mots clés suivants peuvent être utilisés pour définir une couleur : aqua, gray, navy, silver, black, green, olive, teal, blue, lime, purple, white, fuchsia, maroon, red, yellow. Ils ont été choisis pour désigner 16 couleurs bien réparties comme le montre le diagramme suivant :



Le CSS3 les a complétés par 147 mots-clés de couleurs que vous pouvez retrouver à l'adresse <http://www.w3.org/TR/css3-color/#svg-color>.

Vous pouvez être plus précis et accéder aux 16 millions de couleurs que peut afficher un écran en donnant les valeurs R, V et B en hexadécimal des composantes Rouge, Vert et Bleu respectivement de la couleur. Cela s'écrit avec le format #RRVVBB (ou #RVB qui est un raccourci). Par exemple :

- #000000 est noir ;
- #FFFFFF est blanc ;
- #F00, qui est un raccourci pour #FF0000, est rouge ;
- #FF00FF est rose, que l'on peut aussi accéder avec le mot clé pink.

Remarque : vous pouvez utiliser le site Web <https://www.webpagefx.com/web-design/color-picker/> pour trouver facilement le code #RRVVBB de n'importe quelle couleur.

Certains éléments peuvent avoir une taille définie par CSS, d'autres épousent la place minimale nécessaire à leur rendu.

L'unité absolue de mesure la plus utilisée est le « pixel CSS » px. Cette unité correspond souvent avec les pixels physiques de l'écran, mais avec certaines subtilités :

- Les navigateurs actuels permettent de zoomer, ce qui a pour effet d'afficher un « pixel CSS » sur plusieurs pixels ;
- Certains médias comme l'imprimante ne disposent pas de pixels ; le pixel CSS est alors défini comme une échelle permettant une bonne lisibilité sur le média.

Les dimensions sont utilisées sur différentes parties d'un élément. Par exemple, les propriétés CSS `width` et `height` permettent de forcer respectivement la largeur et la hauteur d'un élément (ici les balises `<img>`) :

```
img {  
  width:50px;  
  height:100px;  
}
```

On peut aussi donner des dimensions relatives en pourcentages ; `width:50%` va diminuer de moitié la largeur par rapport à celle qui aurait été calculée normalement.

Référence : pour plus de détails sur les unités disponibles <http://www.w3.org/TR/css3-values/#absolute-lengths>

Nous allons lister ici les propriétés les plus utilisées sur les fontes :

1. `font-size` : cette propriété permet de définir la taille de la police. Exemple :
`font-size:12px;`
2. `font-weight` : cette propriété permet de passer en mode **gras**. Exemple :
`font-weight:bold;`
3. `font-style` : cette propriété permet de définir le style de la fonte (*i.e.* italique ou non). Exemple :
`font-style:italic;`
4. `font-family` : cette propriété permet de choisir la fonte que vous souhaitez utiliser. Exemple :
`font-family: "Lucida Sans Unicode", "Arial", "sans-serif";`

Important : les deux dernières fontes précisées par la règle sont des fontes de secours (fallback) : elles seront utilisées si et seulement si les précédentes ne sont pas disponibles sur le navigateur. Quelques fontes classiques sont répertoriées sur http://www.w3schools.com/cssref/css_websafe_fonts.asp.

Importer de nouvelles fontes : vous pouvez associer à votre page Web de nouvelles fontes à l'aide de la règle `@font-face`. Exemple :

```
@font-face {  
  font-family: myFontName;  
  src: url(path/to/font/font.otf);  
}
```

Voici deux sites pratiques pour télécharger de nouvelles fontes : <http://www.1001fonts.com> et <http://www.fontsquirrel.com>.

Référence : <http://www.w3.org/TR/CSS21/fonts.html> et <http://www.w3.org/TR/css-fonts-3/>.

Nous allons lister ici les propriétés les plus utilisées concernant l'affichage des paragraphes de texte :

1. `text-align` : cette propriété affecte l'alignement des lignes de texte. Exemple :
`text-align:center; /* ou left, right, justify */`

Pour rappel, un paragraphe justifié est un paragraphe où les lignes s'arrêtent à la marge à droite et à gauche ;

2. `line-height` : cette propriété permet d'espacer verticalement les lignes de texte.
Exemple :
`line-height:150%;`
3. `text-indent` : cette propriété indente la première ligne du texte, c'est-à-dire qu'elle la décale horizontalement. Exemple :
`text-indent:12px;`

Si l'on donne un pourcentage comme valeur de `text-indent`, celui-ci est compris comme un pourcentage de la largeur de l'élément parent.

Couleurs : le fond de notre page est tout blanc par défaut. Nous allons changer cela en donnant au `<body>` la couleur qu'a choisie le graphiste/Web-designer : `#838892`.

Dimensions : plusieurs études (cf. 1 et 2) suggèrent que des lignes trop longues ou trop courtes nuisent gravement à la lisibilité d'un site. Pour traiter grossièrement le problème, limitez la largeur de l'élément `<body>` à 600px.

1. Les titres des sections doivent avoir leur texte centré.
2. Le texte doit être aéré : utiliser une hauteur de ligne de 150%.
3. Chaque paragraphe doit être indenté de 5px.
4. Allez chercher une fonte de votre choix sur <http://www.fontsquirrel.com>. Liez-la à votre document avec la règle `@font-face`. Appliquez-la aux titres de section `<h2>` en n'oubliant pas de mettre des fontes en *fallback* (fonte de secours).
Attention : ne mettez pas d'espace dans le nom de votre fonte, ou sinon entourez-la avec des guillemets.
5. Le CSS est un standard au même titre que le HTML. Testez la conformité de votre fichier CSS avec le validateur <https://jigsaw.w3.org/css-validator>.

Les sélecteurs CSS permettent de préciser les éléments qui vont être impactés par la règle CSS. Ils sont aussi utilisés sur d'autres problématiques du développement Web que nous verrons l'année prochaine. Bref vous en aurez au partiel, c'est sûr.

Comme vous le savez, une balise peut prendre des attributs. Deux attributs sont très importants pour les règles CSS : l'identifiant `id` et la classe `class` d'une balise.

Par exemple :

```
<h2 id="monidentifiant" class="skill feature">...</h2>
```

Ce code HTML déclare une balise HTML `<h2>` avec comme identifiant unique `monidentifiant` et ayant deux classes : `skill` et `feature`. **Attention**, un identifiant est unique pour toute la page HTML. Un élément peut avoir plusieurs classes comme dans l'exemple précédent et ces classes sont faites pour être attribuées à de multiples éléments de la page.

Les classes, identifiant et le type des balises permettent de construire 95 % des sélecteurs CSS de base. Voyons la syntaxe pour les utiliser. Il s'agit juste d'utiliser le nom de la balise (`a`, `p`, `img`...) sans autre décorateur. Si on veut donner la couleur rose à tous les liens d'une page, il faut écrire

```
a {  
  color: purple ;  
}
```

Le décorateur associé à l'identifiant est le caractère '#'. Si on veut donner une largeur de 100px à l'unique balise d'identifiant monidentifiant, il faut écrire

```
#monidentifiant {  
    width: 100px;  
}
```

Le décorateur associé aux classes est le caractère '.'. Si on veut donner une hauteur de 200px à tous les éléments qui ont la classe skill, il faut écrire

```
.skill {  
    height: 200px;  
}
```

Tout va principalement se passer dans styles.css.

1. Conformément à la maquette du designer **target.png**, il faut alterner comme couleur de fond pour les titres des sections les valeurs **#5BBDBF** et **#FF5850**. Pour cela il nous faudra ajouter une classe « pair » et « impair » aux éléments **h2** et **h3** et leur associer le style adéquat dans styles.css.
2. L'image **beware.jpg** a du style, mais elle prend un peu trop de place : limitez sa hauteur à 300px.
Attention : la limite de 300px doit s'appliquer seulement à l'image **beware.jpg** et pas à **chuck-jeune.jpg**. Comment faire ?
3. On veut mettre en avant les innombrables arts martiaux que maîtrise Chuck Norris. Pour ce faire :
 - i. On va entourer chacun de ces arts martiaux (Taekwondo, Ju-Jitsu...) d'une balise **** avec la classe **skill** dans le fichier HTML ;
 - ii. D'un autre côté, il faut créer dans le CSS la règle qui associe à tous les éléments ayant la classe **skill** la mise en page suivante : texte en rouge et en italique (ou ce qui vous fait plaisir).
4. Testez la conformité de votre fichier CSS avec le validateur **<https://jigsaw.w3.org/css-validator>**.

Il y a une séparation claire entre les rôles du HTML (contenu avec des balises pour donner du sens) et des CSS (présentation/mise en page). Ce choix n'est pas évident au premier abord : par exemple votre document « Word » ne sépare pas la présentation du contenu. Mais cette séparation est indispensable et très puissante :

- elle permet de réutiliser une présentation d'une page à l'autre. Par exemple quand *lemonde.fr* publie un nouvel article, il ne refait pas le style expressément pour ce dernier: il s'agit d'un nouveau document HTML partageant le même CSS que les articles précédents ;
- elle permet de refaire un site Web en se concentrant sur les CSS sans (trop) toucher au HTML ;
- elle permet de changer la présentation d'un document suivant s'il est destiné à l'impression ou à être visualisé avec un navigateur.